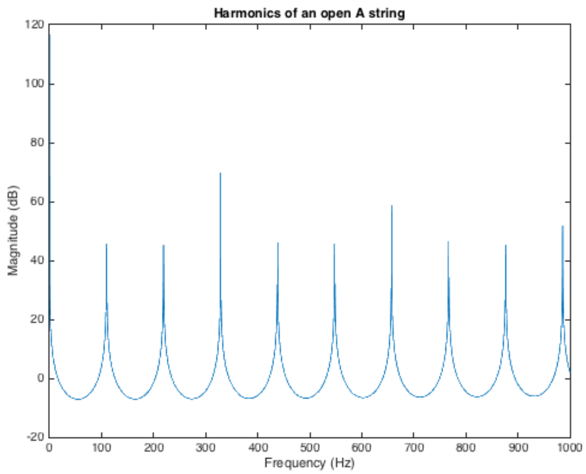
% In-Lecture Assignment #4 on December 2, 2019

% The Karplus-Strong Algorithm synthesizes sounds from stringed instruments.
% See the block diagram and description at
% https://en.wikipedia.org/wiki/Karplus%E2%80%93Strong_string_synthesis
%
% A string in a stringed instrument is anchored on both ends and then plucked
% or strummed. The resulting vibration creates standing waves at a fundamental
% frequency (proportional to 1 over the string length) and its harmonics on the
% string. Musicians can place their fingers to shorten the effective string length
% and change the note (fundamental frequency) being played.
%
% The vibration of the string in turn vibrates the rest of the instrument (e.g. neck
% and body of a guitar) which in turn emits the sound waves that we hear.
% Here's a demo of 2-D standing waves in the body of an acoustic guitar:
% https://www.acs.psu.edu/drussell/guitars.html
% The note and its harmonics form a signature of an instrument. It's how we
% can recognize the instrument as an acoustic guitar, or acoustic bass, etc.
%
% The Karplus-Strong Algorithm consists of an input (excitation) signal and a filter.
% (a) Input signal is a short burst that contains a full spectrum of frequencies.
% (b) The infinite impulse response (IIR) filter passes the fundamental frequency
% and its harmonics from the excitation signal.
%
% Run the Matlab code below.
% 1. Describe the input (excitation) signal.
% *Answer: Short burst to excite the feedback loop. The burst consists of all*
% *discrete-time frequencies so as to allow the filter to pull out the note frequency*
% *and its harmonics. The excitation signal could be a random, pseudo-noise, or*
% *chirp sequence. Here, it's a random sequence generated from a uniform*
% *distribution of numbers on the interval [0,1].*
% 2. Describe the filtering
% *Answer: Due to the feedback loop, the overall filtering from input to output*
% *is an infinite impulse response (IIR) filter, even if the filter in the feedback*
% *loop is an finite impulse response (FIR) as it was in the original Karplus-Strong*
% *implementation. In the Matlab code, an IIR comb filter is used to pass the*
% *the note frequency and its harmonics. The zero locations are altered to*
% *provide additional effects—when removed, it was difficult for me to hear*
% *the difference. The pole-zero plot is at the end.*
% 3. Change the fundamental frequency from 110 Hz ('A' note in second octave
% of the Western scale) and 220 Hz ('A' note in third octave). Describe the sound.
% *Answer: 'A' note at 220 Hz that dampens out with time.*

```matlab
% Karplus-Strong Algorithm from MATLAB help page
% "Generating Guitar Chords using the Karplus-
%  Strong Algorithm".  Edited to play one note.
Fs        = 44100;  % Sampling rate
A         =   110;  % Guitar A string tuned@110 Hz

% Frequency vector for analysis
F = linspace(1/Fs, 1000, 2^12);

% Generate 4s of zeros to generate guitar notes
x = zeros(Fs*4, 1);

% Playing a Note on an Open String
% When a guitar string is plucked or strummed,
% it produces a sound wave with a fundamental
% frequency and harmonics. Harmonics give a
% full sound.  The vibrating string leads to
% standing wave equation because both ends of
% the string are fixed in their location.

% Determine feedback delay based on fundamental
% frequency.
delay = round(Fs/A);

% Generate IIR filter w/ poles approximating
% harmonics of the A string. The zeros are
% added for subtle frequency domain shaping.
b  = firls(42, [0 1/delay 2/delay 1], [0 0 1 1]);
a  = [1 zeros(1, delay) -0.5 -0.5];

% Show the magnitude response of the filter.
[H,W] = freqz(b, a, F, Fs);
plot(W, 20*log10(abs(H)));
title('Harmonics of an open A string');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');

% To generate a 4 second synthetic note,
% first we create a vector of states with
% random numbers. Then we filter zero samples
% using these initial states. This forces
% the random states to exit the filter
% shaped into the harmonics.
zi = rand(max(length(b),length(a))-1,1);
note = filter(b, a, x, zi);

% Normalize the sound for the audioplayer.
note = note - mean(note);
note = note/max(abs(note));
hplayer = audioplayer(note, Fs);
play(hplayer);
```

% Plot of the frequency response of the filter
% It is in the shape of a comb (as in hair comb) with the tines pointing up
% The tines (peaks) are at the fundamental frequency (110 Hz) and its harmonics.

% There is also a strong response at 0 Hz, but we can’t hear that frequency.



Pole-zero plot of comb filter. The ring of poles are just inside the unit circle.